

YATAYATA

Yet Another Toolbar And You AUTO TryIT Anyhow

MULTI\_BAR: -----

Starting Code taken from the internet Forum - Author HEST out of Denmark

Various Functions and Code also from the AUTOIT community - Credit given when possible. Thanks to the whole community for many ideas and apologize for any not given credit.

IMPLEMENTING MULTI\_BAR: -----

\* Create a Folder that will hold all MULTI\_BAR CODE and TOOLBARs. This is logically a TOOLBOX

Unzip MULTI\_BAR.zip in this TOOLBOX Folder

The TOOLBOX Folder will contain:

- \* All user created TOOLBAR INI Files and ICONS

A sub folder TOOLBOX\MULTIBAR for all supplied code will be created containing:

- \* All source code - naming is preceeded by "\$" ( I know - Special Characters, change if you dislike)

- \* All Created Object code is preceeded by "["

- \* All aucillary objects preceeded by "]"

A sub folder of TOOLBOX\SHORTCUTS is optional but was created.

- \* It isolates Shortcuts and any shortcut icons. Many system functions can only be accessed by Shortcuts. Internet Shortcuts are also easily Dragged from Browser and isolated to this folder. Some icons can be retrieved from lnk format, and if not, user icons for the Shortcut can be placed here with the shortcut.

\* The Multi\_Bar Functions (MULTIBAR\$MULTI\_BAR\_Fcns.au3) is ASSUMMED to be placed in the users AUTOIT Includes folder.

If not, it should be EDIT inserted into the \$MULTI\_BAR\_MASTER.au3 source code. Some Functions are from community, eg. "herewasplato", "BugFix", Get\_icon from ?(modified), etc. other techniques from forums and included for ease of compiling.

\* Create the Required "MULTI" BAR Service EXE's using the creation Program

Your TOOLBOX folder\MULTIBAR\$AMultiBARCREATOR.au3. This uses a GUI which will present the supplied defaults and compile those into the Multibar Service Exe's. Defaults are:

- Starting Toolbar Names for each Service position

- X&Y opening position of the FLOATing MULTIBAR

- Users personal ICONs folder path- Optional

When user specific Toolbars are matured, the Creation GUI can be rerun Updating the Starting Toolbars. This is also the case if the X Y position is changed or a user ICON folder is specified.

If CODE changes are made to \$MULTI\_BAR\_MASTER or \$MULTI\_BAR\_fcns , the Creator should be rerun to generate new Exe's.

\* A Callable program is provided to Start or Stop any SERVICE, or Start a TOOLBAR

\Toolbox dir\MULTIBAR\MultibarCTL.au3([barname | servicename,[ servicename | STOP]])

MultibarCTL.au3 provides a callable routine to start any Multibar Float or Edge bar.

It will terminate the running MULTIBAR for that Service position and start the Called Bar

Independant applications can call this routine to start their unique application Toolbar

\Toolbox dir\MULTIBAR\MultibarCTL.au3([barname | servicename,[ servicename | STOP]])

Can Be Used to Start a Service or any TOOLBAR at any service position, or to Stop a Service

ShellExec(MultibarCTL.au3,ARG1 ARG2)

ARGUMENTS can be NO ARGUMENTS, One Argument or TWO Arguments  
consisting of a BARNAME or a SERVICE name

BARNAME = Users Valid XXXXXX\_BAR Name exe

SERVICE Default is "MULTI" or optionally "SIDEL","TOPLT","TOPRT","SIDER"

MultibarCTL.au3 DEFAULT FORMAT/ACTIONS:

1) NO PARAMETERS

ShellExec(MultibarCTL.au3) - Starts MULTI Service with it's Default Starting Bar

2) ONE PARAMETER = Either a BARNAME or a SERVICE name

ShellExec(MultibarCTL.au3, BARNAME) - Starts MULTI Service with BARNAME

-or-

ShellExec(MultibarCTL.au3, SERVICE) - Starts Requested SERVICE with it's Default Starting Bar

3) TWO PARAMTERS

ShellExec(MultibarCTL.au3,BARNAME SERVICE) - Starts Requested SERVICE with BARNAME

- or -

ShellExec(MultibarCTL.au3,SERVICE STOP) - Stops Requested SERVICE

"STOP" is a literal command argument to Stop a Service

Examples can be seen in samples:

\$MULTI\_BAR\_START\_EDGES.au3

&

\$MULTI\_BAR\_STOP\_EDGES.au3

NOTE: Any of the 5 MultiBar Positional .EXE's produced by the Create Program can be executed directly which will present the default Starting Toolbar

\* BUTTONS are defined by dropping a file from the explorer to the button. The Label and Tip for the Button are extracted from FILENAME as well. ICONS are the visual presence of the Buttons and are derived as follows:

1) An ICON with the same Path Filename with a forced ".ico" extensions is searched for in the cons for the Button are retrieved from the same path as the file, with the same name as FILENAME with a .ico extension.

2) If the SameName - file path name.ico ICON file is not found, the search moves on to the Users common ICON folder if specified

3) If still not found, The TOOLBOX folder will be searched for SameName.ico file.

4)If not, the \_GET\_ICON() function will try to obtain the icon from the register or exe, etc.

5)If all fails, a null icon will be shown.

As ICONS are the visual representation of Buttons, the SameName basic logic makes it easy for icons to be created by the user.

## MULTI\_BAR Features: -----

- \* Floating MULTIBAR Toolbar with four(4) Docking Positions
- \* Drag MULTI bar to Dock at any Edge position
- \* Drag Edge bar to screen center to Float as a MULTI bar
- \* FADING EDGE BARS for LEFT, TOP LEFT, TOP RIGHT, RIGHT Sides
- \* All 4 EDGE BARS and MULTI BAR can exist and execute at one time
- \* All Toolbars use common INI file format
- \* Any Toolbar can be displayed by any positional Service EXE
- \* All ToolBars have common Controls
  - Return to calling BAR [ORIGIN}
  - Manual Edit the INI file [INIEDIT]
  - Create and place a NEW Toolbar link on a button [NEWBAR]
  - Search Icon Initiator - search for a Toolbar or a Button Function
  - Set AUTO mode for EDGE bar show on cursor or click, [AUTOSW]
  - Set AUTO mode for FLOAT bar to close or stay open on button click [AUTOSW]
  - Rotate thru 3 button sizes, small, medium and large w/label [B-SIZE]
  - EXIT this bar [EXIT]
- \* User specifies Number of BUTTONS and Number of ROWS
- \* Extend Dynamically Number of BUTTONS or ROW
  - via NEWBAR Function Specifying SAME BARNAME with Changed BUTTONS & ROWS
- \* Three(3) Button sizes - User can change on demand
- \* Shrink to Fit - WILL Auto reduce Button size on DOCKING if Bar too long
- \* Buttons can be any File OPEN function or Open another TOOLBAR\_BAR
- \* ToolBars can be cascaded down(DRILL DOWN-Button points to another Toolbar) to additional Toolbars with Return to previous Toolbar via Origin Function
- \* 2 BAR TYPES:
  - ACTION(Buttons do TOOLBAR OR File OPENS)
    - User drops a Link on Button
  - DROP(Buttons are Folder Targets)dropped files are sorted to destination
    - MOVE or COPY: FILE, FILE(s) or FOLDERS(DIRs) to Button target
    - FILES OR FOLDERS CAN BE DROPPED TO A RECYCLE BIN BUTTON
- \* Button context functions
  - DELETE the current function - empty the button
  - EDIT the TOOLTIP for this Button
  - EDIT the LABEL displayed on Large Button
  - PLUCK this Button for move to new location on this bar -or- ANY other Bar (in this TOOLBOX)
  - PLACE any PLUCKED Button, or PLACE any NEW TOOL\_BAR
  - OPEN file location of Button file

The main folder containing the \MULTIBAR subfolder with the MULTI\_BAR generating code and resulting Bar Executables, as well as all toolbar INI's is considered a TOOLBOX. This common location permits the movement of buttons between toolbars as well as displaying any toolbar (?\_BAR.ini) in any toolbar location by any of the EXE's. It can also contain any unique user file icons.

This TOOLBOX folder will contain all the user defined TOOLBAR definitions in the form of BarName\_BAR.ini files. It is also the first folder searched for ICONS that match the BarName or match the file link name eg. BarName.ico or Anyfilelink.ico . There is a provision, if the user has a local cache of personal icons, to specify their icons folder, which is searched second for icons if specified. If an icon is not then found , a search is made of the ToolBox folder and if not found there a possible icon from Register based on file type or extracted from exe or .lnk structure. This code to extract icons is short on function, hence user created or renamed icons can be used.

DELIVERABLES: -----

- 1) \$MULTI\_BAR\_MASTER.au3 - This is the source for all Toolbar Positions. It need not be modified unless additional functions beyond what is provided are desired.
- 2) \$MULTI\_BAR\_fcns - This is a collection of Functions and should be placed in the users AUTOit INCLUDE location, but can be appended by edit to MASTER if desired.
- 3) \$AMultiBARCREATOR.au3 - This Program will create Positional executables. It presents a GUI Option screen to collect user OPTIONS:  
For each POSITIONAL EXE: MULTIBAR(FLOAT) SIDEL, TOPLT, TOPRT, SIDER  
\$StartingBARNAME = This is the NAME of the FIRST or INITIAL BAR shown for this position.  
It requires the form "xxxxx\_BAR". It is represented in file as "xxxxx\_BAR.ini"  
A Default Starting bar is initially provided  
When the user creates a Specific NewBar, he can update this as Default  
\$iFloatX - IF a MULTIBAR FLOAT BAR is being created this is initial x and y coordinates  
\$iFloatY  
\$USERIconsPath = Path to a optional and additional Icon folder. The TOOLBOX or script path will be searched first for user icons, and if the user has a common store for icons this path ending in "\" will be searched. There is a get icon function that attempts to get an icon for all file types, but
  - 1) It is imperfect, and
  - 2) user can supply his own icon for any file
- 4) \$MULTI\_BAR\_MASTER.ini - Retains any UpDated Variables from \$AMultiBARCREATOR.au3 and the initialized variables provided.
- 5) \$Multi\_BUTTONSEARCH.au3- An auxillary program which is invoked from the Search Icon, or can also be assigned to a Button. Given a Text String, will search the INI files of the TOOLBOX for a "\_BAR" name and open that Toolbar , or search each Button Function and present the containing Toolbar for a short time period until USER confirms this Bar contains the Function being sought, until all Toolbar INI Files have been searched.
- 6) \$MULTI\_BARPluckTimeOut.au3 - This routine provides an abort function when moving a button(Pluck/Place) from one Toolbar to another. Buttons can be moved within a single Toolbar, another Toolbar cascaded from the Origin Toolbar, or from a Toolbar at a given Position to a different Positional Toolbar. If not completed in a reasonable time. eg From a SIDEL Toolbar Button to a TOPRT Toolbar Button, The Move is aborted.
- 7) MultibarCTL.au3([[barname | servicename], servicename | STOP]  
A user callable routine to start a toolbar and or service, or Stop a Service  
Defaults and actions:  
No parameters - Starts Multi Float with default Starting Bar  
Barname - Starts Specific BarName as Float  
Servicename - Starts Specific Service with Default Starting Bar  
Barname, Servicename - Starts Specific Barname on Specific Service  
Servicename, STOP - Stops Specific Service
- 8) HOME\_BAR.ini, STARTINGxx\_BAR.ini - Default Starting bars - In TOOLBOX folder
- 9) \$Multi\_EDGE\_xxxx.ico An Icon file for EDGE, Float and Master EXE's
- 10) ]BUTTONSEARCHHIT.wav - Sound File for Button Search Hit

11) ]README.doc - or - [README.pdf = This text

12) TOOLBAR Controls ICONS-]Origin,]INIEDIT,]NEWBAR,]ButtonSearch,]AUTOSW,]BSIZE

13) SAMPLEs - use of MultibarCTL.au3 to Start and Stop Services and Start Toolbars

\$MULTI\_BAR\_START\_EDGES.au3

\$MULTI\_BAR\_STOP\_EDGES.au3

14) Default ini and Icons

## ABOUT MULTI\_BAR: -----

MULTI\_BAR is offered as a single source (\$Multi\_Bar\_Master.au3) that supports 4 fixed positional EDGE ToolBars and a Floating toolbar with Service EXE's for each position.

User Options for these are Selected via a GUI which modifies the source as a precompile processor to create a temporary au3 source which is compiled into a Service executable which supports each of the fixed EDGE bar positions and the FLOAT bar.

There is a requirement that the FLOAT service routine which is named [MULTI\_BAR.exe and all Edge Service Routines be Created. The User is not required to utilize all positions, as they may choose to use only one or more EDGE toolbars. This FLOAT toolbar does support the Button Search Routine for displaying successive toolbars with any button hit on the search string. The user then indicates if this is the Toolbar he is looking for. Edge bar Service will be activated on any DragNDrop to that position.

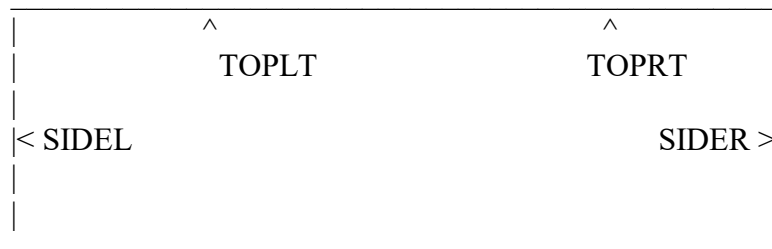
In General, because of the cascading provision of toolbar-in-toolbar, many Toolbars can be accessed from one Service Routine be it FLOAT or EDGE bar Position. Toolbar Positions are more a function of user preference for mouse activation.

---

### Multi\_Bar Code

\* Can be used to produce any one of the FOUR (4) Fixed EDGE Bars based on Screen Positions:

- 1) SIDEL
- 2) TOPLT
- 3) TOPRT
- 4) SIDER



`The Executable for each Position has a Fixed EXE NAME, but the initial TOOLBAR NAME is optional

\* It also creates a FLOAT(ing) BAR executable [MULTIBAR.exe

1- The user will specify the default Starting Bar for each positional BAR service executable:  
"MULTI","SIDEL","TOPLT","TOPRT","SIDER"

2- The \$MULTI\_BAR\_MASTER.au3 source will be read and modified

3- The Modified Source will be saved as a temporary source file which is compiled into an executable:  
EDGE BAR EXEs are FIXED NAME:  
[SIDEL\_BAR.exe,[TOPLT\_BAR.exe, [TOPRT\_BAR.exe, [SIDER\_BAR.exe  
One FLOAT BAR exe name [MULTI\_BAR.exe is required

The main program [MULTIBAR.exe supports a Button Search Program to find if a button exist on any Toolbar as toolbars proliferate. Any hits will be presented as a Floating Toolbar

Each TOOLBAR is named xxxxx\_BAR. the UnderscoreBAR is required

This is the TOOLBAR\_BAR.ini definition file.



THE FOLLOWING BAR ICON CONTROLS EXIST: -----

- \* ORIGIN - Return to calling TOOLBAR
- \* INEDIT - Edit the TOOLBAR INI file directly
- \* NEWBAR - Create a new TOOLBAR  
????? \_BAR,NumButtons[15],NumRows[1],Type[ACTION](Default) or DROP  
PARAMETERS:
  - ????? \_BAR - Must end in "\_BAR" - Result will be ?????? \_BAR.ini file
  - NumButtons[15] - Not program limited, but may exceed screen size - Control to Exit may be unaccessible
  - NumRows[1] - Not program limited, but may be impractical
  - Type[ACTION](Default) or DROP - Drop is used to sort files into drop boxes(Folders)
- An BAR .icon FILE with the same Name ????? \_BAR.ico will be displayed if available
- \* SEARCH - Search for a TOOLBAR Name or a STRING in any Button on all TOOLBARS
  - If String does not end with ????? \_BAR the string is assumed to be a button search string.
  - \$Mult\_ButtonSearch.au3 is invoked.
- \* AUTOSW (Toggle Switch) Control TOOLBAR viewing actions  
DEPENDING ON TOOLBAR
  - IF "FLOAT" - Toggles whether Toolbar Stays open after selection  
or Closes after selection(default)
  - IF "EDGE" -Toggles whether Toolbar Unhides from Cursor  
or Unhides from MouseClick
- \* B-SIZE - Dynamically Changes BUTTON Size to one of three Sizes
  - Small - Icon only
  - Medium - Icon only
  - Large - Shows Icon and a LABEL



## INI FILE FORMATS: -----

Sample INI FILE FORMAT = NEWBARS Created with these default entries

```
[BAR]
    NAME=YOURNAMED_BAR
    TYPE=ACTION
    BUTTON_NUM=10
    BUTTON_ROW=1
    BSIZE=2

[Launch]
    1=@ScriptDir
    5=@ScriptDir\ $MULTI_BUTTONSEARCH.au
    10=@ScriptDir\]README].pdf

[Params]
    1=
    5=
    10=

[ToolTip]
    1=TOOLBOX_DIR
    5=BAR & BUTTON_SEARCH
    10=README

[Label]
    1=_____
    5=_____
    10=_____

[AUTOMODE]
    AUTOSW=1
```

## BUTTON CONTEXT MENU: -----

- \* Delete Function - clear button
- \* EDIT ToolTip
- \* EDIT Button Label
  - Change Default Button Label - shown only on Button Size 3
- \* PLUCK Button for Move - Button can be moved within
  - or across FLOAT TOOLBARS
  - or across SAME EDGE TOOLBARS
  - Sequence:
    - Pluck Button
    - USER Selects the new toolbar
    - PLACE on button
- \* PLACE Button Moves, Newbars
  - Target for Buttons Moves
  - Target for NEWBAR FUNCTION just defined
- \* OPEN LOCATION -> Go to File Location
- \* EDIT File - starts defined EDIT function for BUTTON File type